International Journal of Parallel, Emergent and Distributed Systems

ISSN: 1744-5760 (Print) 1744-5779 (Online) Journal homepage: https://www.tandfonline.com/loi/gpaa20

Fifty years in home computing, the digital computer and its private use(er)s

Stefan Höltgen

To cite this article: Stefan Höltgen (2020) Fifty years in home computing, the digital computer and its private use(er)s, International Journal of Parallel, Emergent and Distributed Systems, 35:2, 170-184, DOI: 10.1080/17445760.2019.1597085

To link to this article: https://doi.org/10.1080/17445760.2019.1597085

Published online: 26 Mar 2019.

INTERNATIONAL JOURNAL OF PARALLEL, EMERGENT AND DISTRIBUTED SYSTEMS
2020, VOL. 35, NO. 2, 170–184
https://doi.org/10.1080/17445760.2019.1597085

Fifty years in home computing, the digital computer and its private use(er)s

Stefan Höltgen

Department for Musicology and Media Science, Humboldt University, Berlin, Germany

ABSTRACT
The following chapter will discuss the relation between home computer history and computer programming – with a focus on game programming. The nurseries of the early 1980s are the origins of the later computer game industry and the private use of microcomputers becomes an essential part of the 'playful' exploration and emancipation of technology. This is why the contribution that home computing added to the history of digital economics, culture and even thought cannot be estimated high enough. After the definition of central terms and discussing the main problems of the historiography of technology and computing, the main topic separated in decades will be portrayed. The milestones of home computer technology will be linked to the scenes and (sub)cultures of their users. Some of these links will be highlighted. My presentation will not be exhaustive but will try to highlight some of the playful uses of computers – from the perspective of this technology.

ARTICLE HISTORY
Received 13 March 2019
Accepted 16 March 2019

KEYWORDS
Homecomputer; computer history; video games; hacking; BASIC

SUBJECT CLASSIFICATION CODES
Include these here if the journal requires them

Writing about the history of home computing – even as an overview like here – will be faced with some issues. Initially, the term 'home computer' or even 'computer' has to be appropriately defined to isolate the subject: does the term computer cover analogical as well as digital and mechanical as well as electronic calculating machines? Above all: what are the attributes of a home computer? Is it that it is not restricted to economic, scientifical, or military areas? Does it consist of specific features that make it 'homelike'? Or is it its price, its serial production, and general availability?

Another problem of epistemological nature joins this area. Writing a 'historiography' of home computers forces this technology into a very anthropomorphic scheme: a classification by eras/epochs/decades/ . . . , linking the technology to biographies (e.g. of inventors or companies), and even looking at it from the perspective of their users would be an alien approach to the technology itself. This alienation can not only be seen by comparing the human and the machine's time (when the clock speeds of digital computers elude the human time perception) but also within the anachronisms of the machine as a combination of different components from different times (both of their invention and their construction). Finally, the time problems escalate when the 'historical' home computer is switched on and calculates problems not of 'its time' but of the present age – like contemporary games.

I prepended with these considerations to sensitise the reader for the problems and contradictions of historicising computers. It should be noticed that my contribution does not try to tell a techno-historiography of home *computers*[1] but the history of home *computing*. The difference is of vital importance since such a history will ask the following questions: What had been done (in the sense of

CONTACT Stefan Höltgen stefan.hoeltgen@hu-berlin.de Department for Musicology and Media Science, Humboldt University, Georgenstraße 47, D-10117, Berlin, Germany

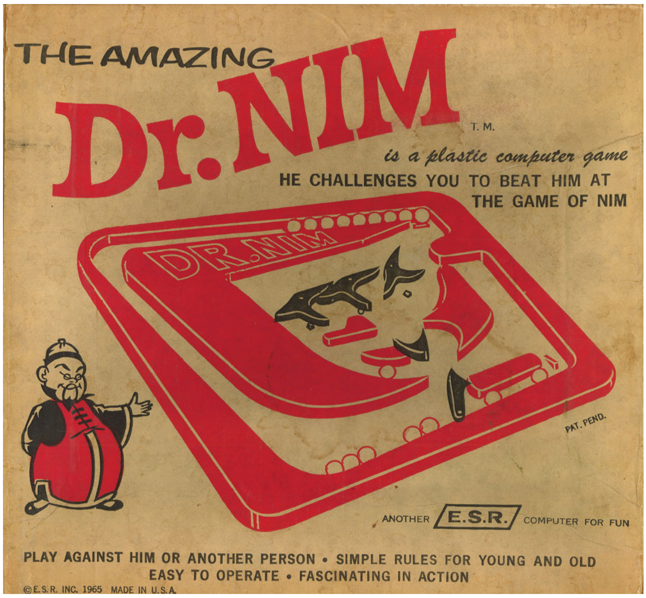

Figure 1. Dr. NIM – a 'plastic computer' from 1964.

creative human activity) with computers in the 'homes' back in the day? And: What did (and do) home computers themselves do (in the sense of technical operation as evoked by the present progressive form '-ing')? Here the history of technology and the histories of the society, culture, and thought are connected with each other.

1. Pre-History

The private use of computers did not start when people were able to buy digital computers in stores. Its origins can be found in the 'misuse' of such technology for private purposes by their users. Here the primal scene of hacking in the 1950s at MIT (Boston) [1, p.1–26] or the use of an IBM 650 at Stanford University by students who calculated pairings with students and nurses (from a nearby nurses' dormitory) in the computer centre at night [4, p.274ff.] could be mentioned. Above this, the tradition of 'calculating toys' should be mentioned since these toys are an important predecessor of computers, too: the marble computer for kids, constructions kits, board games and toys with calculating and computing capabilities as the 'Dr. NIM' and 'Digi-Comp' board games published in the 1960s by E.S.R. Inc (Figure 1).[2]

Thus home computing looks back on a pre-history of calculating hard- and also software that had been adapted for the use of non-scientific users. First of all, there are programming languages like BASIC (1964, Dartmouth College) or LOGO (three years later at BBN in Cambridge, Massachusetts). Both languages had been adapted many times to different computer platforms and are used even today in teaching and retrocomputing.

The miniaturisation of electronic technology also shrunk computer components (e.g. from mechanical switches, to relays, valves, and transistors) and had an influence on the production costs, energy consumption, and prices. Finally, in the late 1960s/early 1970s, when the TTL technology[3] made computers even more compact, mass production could start. That decreased prices to a low five-figure number which enabled wealthy private users to afford a mini computer. Exotic machines like the Honeywell H316 that had been sold from 1969 as a 'Kitchen Computer'[4] or the PDP-8 from Digital Equipment Corporation (DEC) that hit the market in 1965 (and had been sold more than 50.000 times until its production had been stopped in 1979) are examples for that (Figure 2).

Figure 2. PDP-8/e – one of the first affordable minicomputer.

Figure 3. So-20 – Keyboard computer with all necessary components in one case.

In 1969 the H316 rarely had been used as a storage system for receipts in kitchens. But in the same year, Lee Felsenstein and some of his hackers friends got hold of an outdated SDS-940 mainframe computer from the University of Berkeley. They used it and its time-sharing operating system[5] for building the probably first social network in the San Francisco Bay area: Connected via landline to several terminal systems that had been placed in public spaces (like record stores) everybody could connect to the computer and use it as a bulletin board or to chat with other users. [cf. 4, p.385–404] Privatisation of computer technology became the main project for Felsenstein and his colleagues. Later he invented two of the first all-in-one computers (the Sol-20) and luggable computers (the Osborne-1) and became a member of the Homebrew Computer Club. Founded in 1975 this club spread the ethics of the MIT hackers into the public discourse: a call for data privacy, freedom of information, and for the emancipated and creative use of computer technology [cf. [1, p.26–36] (Figure 3).

Finally, computer games played a leading role in the privatisation of computer technology. The constituted a kind of 'interface' between the high technology and the desires and abilities of people that had nothing to do with computers until then. The history and genealogy of video games have often been discussed; here I'd like to mention three impulsions of computer games and home computing: 1. computer games drove the development of specified sound and graphics hardware and programming languages (to programme that hardware) [cf. 5], 2. the provoked early activities with computers (especially when it comes to programming), and 3. they co-initiated the home computer era in the 1970s.

2. The 70s

The last point should be specified: the CPU as an integrated circuit has been introduced in the autumn of 1971 by Intel. Until 1975 Intel developed this technology up to an 8-bit architecture and other companies competed with their own CPUs: Motorola, Texas Instruments, MOS Technology, Fairchild, RCA, and Zilog built their very own architectures and put them in their computers or sold them to computer manufacturers. Up until that time small computers had been built with TTL technology. Computer games followed the same 'logic': games like 'Pong' (1972) or 'Break-Out' (1975) by Atari were implemented with TTL completely. The use of microprocessors allowed manufacturers to be more flexible in developing games: the game hardware could be designed more 'generally' by using CPUs and the game itself could be programmed in software. This division made it possible to become a game developer without having the knowledge or facilities to produce hardware which brought many new game programmers to the market – mostly 'game hackers' who developed software for arcades or homecomputers on their own computers.

The step by step transformation of computers into gaming machines had its second origin in the minicomputer era of the late 1960s. Games like 'Spacewar!' had been programmed for the PDP-6 by students and DEC used them for advertising and to test the performance of their hardware. It may be that early 'game affinity' of *DEC* that let to the decision of publishing a book on '101 BASIC Games' (collected by David H. Ahl [6]) in 1970. Those BASIC games had to be typed into their computers by the users. The book gained such a huge popularity that Ahl published a sequel [7] and later a 'microcomputer edition' [8]. This was the time when magazines for electronic and computer hobbyists flooded the magazine market: publications with computer kits (e.g. 'Popular Electronics'), club magazines and magazines for specific platforms in different countries. Most of those magazines contained BASIC game programmes that had to be typed in by the readers which was a low price alternative for commercial game software.

The formerly mentioned *Homebrew Computer Club* was an early melting pot for hardware and software tinkerer, microcomputer artists, hacker and techno-political activists (like Ted Nelson who had dedicated his work [9] to the technological enlightenment of the masses). But also companies like *Apple* (who introduced their Apple I computer in 1976 at a club meeting) and *Micro-Soft* (sic) (who tried to sell their Altair 8800 BASIC to the club members) started there. While the Microsoft-BASIC provoked the very first copyright controversy,[6] the Apple inventors used the earnings from their single-board computer to fund a commercial computer platform and explore the market.

The PET 2001 (by Commodore), the TRS-80 (by Radio Shack) and the MZ-80 by Sharp were the competitors of the Apple II in 1977. The four systems were sold in different price classes. The Apple II (a well-known anecdote from its constructor Steve Wozniak says) had been the realisation of a computer game system: Wozniak made the TTL board 'Break-Out' for Atari and wanted to implement the game into a very own computer system. For his 'Break-Out' computer he needed adequate hardware (with graphic and sound capabilities and controller connections) and a programming language. Both were invented 'from scratch': the Apple II and its 'Game BASIC'.[7] The running 'Break-Out'[8] game (Figure 4) was the proof-of-concept.

The game was published on audio (data) tape but could also be typed in from the BASIC listing print-out. Since the first home computers hit the market, nearly all of the machines featured a built-in BASIC programming language, so it wasn't necessary to compile a BASIC source code into machine language to run the programme as with the minicomputer dialects. Because of the minimum of RAM and external slow mass memory (audio tapes, floppies, ...), the BASIC had been implemented into the ROMs of the computer and used an interpreter instead of a compiler. With this, a BASIC source code had to be typed into the shell and was stored tokenised[9] from the interpreter immediately. To run it the interpreter translated the BASIC tokens step by step into machine code in real-time. The advantage of an easy-to-debug code came at the price of very slow performance.

The mentioned home computers disposed of very different architectures and configurations. They used different CPUs (mostly *Zilog's* Z80 or *MOS's* 6502), different amounts of RAM memory, graphic and

Figure 4. Apple II with 'Brick-Out' game in BASIC.

sound capabilities. When programming games the user was confronted with different technological obstacles of those platforms and with the slowness of their BASIC interpreters. Those disadvantages forced some of the hobby programmers to learn assembly languages. Assembly programmes are running much faster than BASIC programmes, use less memory and allowed the direct programming of the computer's hardware elements. To programme with assembly language either a special (commercial) development system had to be used or the programme could have been 'poked' into the computer's memory by using a so-called BASIC loader. With this BASIC was only used for storing the machine language opcodes[10] into the RAM and starting the completed programme afterwards. (Therefore specific commands were implemented in nearly all home computer BASIC versions.) A BASIC routine to carry that out had the following form:

```
10 MEM = 1000
20 READ A
30 IF A = −1 THEN END
40 POKE MEM,A
50 MEM = MEM+1
60 GOTO 20
70 DATA 169,211,174,0,0,157,0,128
80 DATA 157,0,129,157,0,130,157,0
90 DATA 131,232,205,255,0,208,238
100 DATA −1
```
[11]

Typed into the Commodore PET 2001 and started with RUN the programme reads the data from line 70–100 and writes it into the RAM location starting from 1000. After this is done, the machine language programme can be started with the BASIC command SYS 1000 (Figure 5).

With the programme above some didactic 'problems' of BASIC can be demonstrated. Loops could be programmed with GOTO statements since the line numbers worked as labels. It was allowed to break a loop process (here with END) or to jump out of it with GOTO, too. In longer programmes, those practices lead to 'spaghetti code': BASIC programmes that made use of those concepts are hard to understand (for people) and even harder to maintain and debug. Above this the use of the DATA table shows the enormous effort of programming assembly for such computer systems: If the user could not afford an assembler he/she had to translate the mnemonical codes into (here: decimal) opcodes manually by precisely considering the used RAM addresses and addressing modes. A human reader couldn't comprehend the content and function of such a hand-coded assembly language programme.

Figure 5. BASIC loader (left) and execution of the machine language programme (right) on a PET 2001.

Figure 6. The German NDR Klein Computer (a kit computer from a German TV show) was a modular system that could be extended up to a 16-bit computer.

At the end of the 1970s computers had even been recognised by schools as a 'future technology' and entered the curricula of elementary and high schools. In the beginning, the focus was on the comprehension of computer hardware and how software could affect it. Single board computers were mostly used for this teaching: their construction from a kit step-by-step with modules for different functions up to a running 8-bit microcomputer with high level programming languages and peripherals (keyboard, monitor, printer) (Figure 6) As a kind of echo from the era of homebrew computing and its magazines a culture of playfully learning computer science autodidactically established. Even for those self-made systems, magazines and books with type-in programme listings emerged.

3. The 80s

After the first generation of home computers had been displaced with models that contained more RAM memory, the programmes got larger, too. This led to pages full of DATA lists that had to be typed in correctly. (Figure 7) But more RAM was not the only benefit of this new generation of home computers. They were equipped with special processors that could be used developing and playing games.

This trend started when the Video Computer System (VCS) was developed by Atari: as a gaming console it had to be cheap, so its technical equipment was very low-level: a CPU that could only run programmes with four kilobytes, nearly no RAM memory (only 127 bytes of scratchpad RAM) and therefore no video memory. To produce graphics and sounds a new type of co-processor was invented for the VCS by Jay Miner: the Television Interface Adapter (TIA) that could produce the audiovisual output for the games. Shortly after that other competitors developed and published similar chips to put them in their own home computer models or sell them to computer manufacturers. Those different technologies made the computer platforms incompatible to each other. This could be recognised by the different BASIC dialects: each home computer used special commands and functions to utilise the features of its built-in special chips. But even with this technological progress fast-moving graphics and concurrent sounds could only be programmed in assembly languages.

When the first 'video game crash' happened in 1983 most of the game console manufacturers went bankrupt which accelerated the spread of home computers. It was Atari who ignored the technological progress in computer technology and tried to extend the life of its 1976 designed VCS instead of developing a modern system. So the gamers lost their trust in the video game consoles and bought computers for gaming. Home computers not only were better configured with memory, graphics, and sound – they also could be used for being creative with programming. Indeed Atari and other console manufacturers tried to sell computing extensions to make their machines programmable with BASIC and assembly language. (Figure 8) But the resulting programmes couldn't compete with those from home computers [10].

Like the phoenix from its ashes the new 8- and 16-bit video game consoles (from Sega, Nintendo, and others) rose from the ruins the 'video game crash' left. Those new systems were provided with special hardware for graphics and sounds and even specialised game controllers so the home computers could not compete with them anymore. Above this, a professional market for game software emerged in the first half of the 1980s. From now on games were programmed with professional tools from former homebrew programmers that sometimes had become 'stars'. This professionalisation and commercialisation of the game sphere provoked a counterculture: commercial software, especially games, were 'cracked' by gamers with programming knowledge, cheats were added and black copies circulated. The crackers began to add short intros to those games to show who cracked them. Those intros became more complex over the years – in regard to graphics and sound – and established a new art form. Those 'cracktro' led to the demo scenes which are a part of the digital arts until today.

Interesting about those demos was the fact that they lead to competitions amongst the crackers/programmers. Everybody wanted to outdo each other in programming a specific system to conquer its technological constraints. By doing this, new algorithms and programming techniques were discovered and even 'new' hardware specs were detected – e.g. undocumented features that resulted from design flaws. If such flaws could be used to reach a specific effect they were explored and added to the knowledge base of the specific home computer platform. Professional software developers benefited from this knowledge, too, when they used it to add more sound channels, colours, graphical elements or higher screen resolutions to their games. Using those hacks even obsolete platforms like Atari's VCS excelled themselves for decades and until today.[12]

4. The 90s

Not only gaming consoles but also home computers reached the end of their 8-bit era in the middle of the 1980s. When microprocessor manufacturers (*Texas Instruments's* TMS9900, *Intel's* 8086, and *Motorola's* 68000) started to produce and sell 16-bit CPUs in the second half of the 1970s ten years

Sinclair-Praxis

Gedanken-Lese-Spiel

Der Spectrum (16K und 48K) errät in wenigen Versuchen eine Zahl, die Sie sich ausgedacht haben. Wie macht er das wohl?

```
  10 BORDER 3: PAPER 1: INK 6: C
LS
  20 PRINT TAB 3;"****** GEDANKE
NLESEN ******"''TAB 3;"(C) by Ch
ristian Goetz 1984"
  30 FOR n=1 TO 10: BEEP .05,n:
NEXT n
  40 INPUT "Kennst Du das Spiel
?,(J/N) CLS ";a$
  50 RETURN a$="j" THEN GO TO 10
0
  60 PRINT ''"Merke Dir eine Zah
l,die zwischen1 und 63 liegt. De
r Computer     zeigt Dir dann 6 v
erschiedene    Zahlentafeln und f
ragt Dich      lediglich, ob Dein
e Geheimzahl   sich darunter befi
ndet.          Du antwortest jewe
ils mit J(a)   oder N(ein)! Dann
laesst er      seine magische Kra
ft walten..."
  70 PRINT ''TAB 6; FLASH 1;"STA
RT mit Enter !!": PAUSE 0
 100 FOR n=10 TO 1 STEP -1: BEEP
 .05,n: NEXT n: RESTORE
 105 LET z=0
 110 FOR j=1 TO 6: GO SUB 400
 120 INPUT "Steht die von Dir ge
dachte Zahl jetzt auf dem Bildsc
hirm ? (J/N)";a$
 130 IF a$="j" THEN LET z=z+p
 140 NEXT j: CLS : PRINT ''TAB 6
;"Deine Zahl war :"
 145 FOR n=1 TO 10: READ a: BEEP
 .1,a: NEXT n
 150 PRINT 'TAB 6;"**************
***"
 160 PRINT TAB 6;"*
 *"
 170 PRINT TAB 6;"***************
**";AT 5,13; BRIGHT 1; FLASH 1;z
 180 BRIGHT 0: INPUT "Willst Du
es nochmal versuchen ?(J/N)";a$:
 IF a$="j" THEN RUN
 190 PRINT ''"Wer nicht will, de
r hat schon...": STOP
 400 CLS : LET p=0: FOR n=1 TO 4
: FOR f=1 TO 8: READ a: IF p=0 T
HEN LET p=a
 410 PRINT AT 2*n,3*f;a: NEXT f:
 NEXT n
 499 RETURN
 500 DATA 1,3,5,7,9,11,13,15,17,
19,21,23,25,27,29,31,33,35,37,39
,41,43,45,47,49,51,53,55,57,59,6
1,63
 510 DATA 2,3,6,7,10,11,14,15,18
,19,22,23,26,27,30,31,34,35,38,3
9,42,43,46,47,50,51,54,55,58,59,
62,63
 515 DATA 4,5,6,7,12,13,14,15,20
,21,22,23,28,29,30,31,36,37,38,3
9,44,45,46,47,52,53,54,55,60,61,
62,63
 520 DATA 8,9,10,11,12,13,14,15,
24,25,26,27,28,29,30,31,40,41,42
,43,44,45,46,47,56,57,58,59,60,6
1,62,63
 530 DATA 16,17,18,19,20,21,22,2
3,24,25,26,27,28,29,30,31,48,49,
50,51,52,53,54,55,56,57,58,59,60
,61,62,63
 540 DATA 32,33,34,35,36,37,38,3
9,40,41,42,43,44,45,46,47,48,49,
50,51,52,53,54,55,56,57,58,59,60
,61,62,63
 550 DATA 0,4,7,12,7,7,12,7,7,12
```


Nr. 10 – Oktober 1984 HC 93

Figure 7. A page from a German home computer magazine with a part of a programme listing.

Figure 8. Atari VCS with the Spectravideo BASIC extension. The BASIC interpreter resides in the game ROM cartridge the keyboard was attached to the console via its game ports. Programming was just another kind of gaming with such extensions.

later those chips got cheap enough to be implemented in home computers and video game consoles.[13] *Apple's* LISA (1983) and Macintosh (1984) computers, *Sinclair's* QL (1984) *Commodore's* Amiga (1984), and *Atari's* ST (1985) started the 16-bit era in home computing. They featured more RAM capacity, faster and bigger peripheral memory and new special chips for sound, graphics and mathematical operations.

Programming those systems was uncommonly performed with 16-bit assembly languages since they were much more extensive and complicated than those for 8-bit CPUs. Because of their better configuration compiled programming languages like C (and sometimes compiled BASIC dialects) were used for software development. The mentioned systems (with the exception of the QL) possessed operating systems with graphical user interfaces (GUI) that were put between the computer hardware and the users like a 'shield'. (Macintosh computers didn't even come with a BASIC. The programming of the Mac by the user wasn't intended by *Apple*.) The software development for those platforms had changed and professionalised. Integrated Development Environments (IDEs) were used to programme, compile, and debug C source codes comfortably. BASIC Listings for games disappeared from the magazines and books for those home computers.

The emergence of 16-bit computers did not mean the disappearance of the former home computer generation. At the end of the 1980s, an innumerable amount of platforms from different manufacturers existed all over the world. Nearly each of those platforms did create user groups, with specialised knowledge about hardware as well as software and magazines for their system. By accessing this knowledge emulators for 8-bit systems were developed to run on the new 16-bit computers. This process was started in 1985 when Commodore published their last 8-bit computer, the C128, which integrated a full Commodore C64 (so the C64 software, mainly games, could be used with the C128). Likewise, *Sinclair* added downward compatibility to each new Spectrum computer so the software of the 1982 versions could be used with them by just choosing the '48 BASIC' version in the startup menu (Figure 9).

The ZX Spectrum presumably was the first home computer for which a software emulator was developed: in 1991 NUTRIA[14] (Figure 10) was published for MS/DOS on 16-bit IBM compatible computers. Emulators for complex machinery do exist since 1940 and for computers since the 1960s.[15] Those were of professional origin[16] and for commercial use. Emulators for home computer systems are mostly the work of hobbyists who invest their capital (time and knowledge) in a virtual replica of their most loved platform.

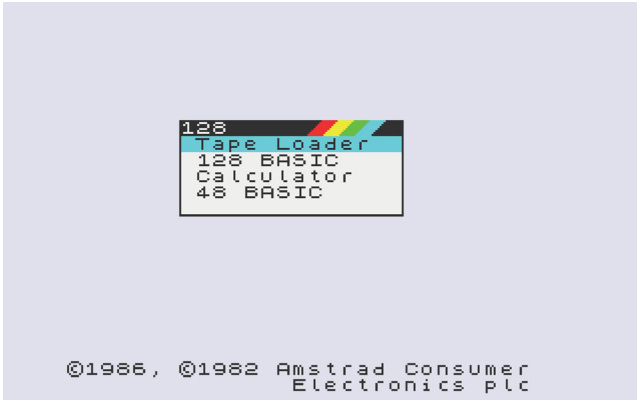

Figure 9. Startup screen of the Spectrum + 2 (1986).

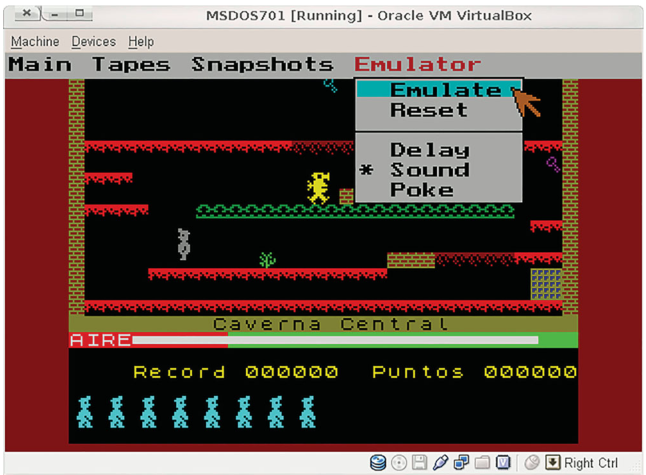

Figure 10. NUTRIA – the first emulator for Sinclair's ZX Spectrum home computer.

5. . . . and beyond

Computer history seems to be preservable and operative with the help of emulators. But emulators differ a lot from the systems they try to clone. Most times they only try to rebuild the outputs of a given computer – not by way of simulating its internal structures but by mimicking their timings, sound, and graphic capabilities. Those kind of emulators are often calibrated by running historical software: Once they are able to run vintage games, they seem to emulate the system accurately. Since they do not emulate hardware specifics, they can't be used for the development and testing of new software for vintage computers. But this is one of the most interesting domains of retro computing.

What does retro computing mean? The prefix 'retro' stems from the Latin and can be translated as 'backwards'. This aiming backwards of retro styles should not be mistaken as conservatism because retro scenes don't try to get back a past status-quo. Retro is more about transferring elements from the past into the present. By combining these two time levels, it forces the observer/user to correlate both levels with each other. Therefore retro computing is even more than getting historical computers back in operation. Even when the user just wants to do the things he did back in the day (when he/she first used the computer, like gaming), he/she will do this here and now. He/she has to consider his/her biography, knowledge and the present times when using an old computer. By this retro computing will become a technological anachronism: it brings computer history up-to-date.

Any (historical or not) computer does this 'update' process by being switched on. In operating status, it is radically present. The premise for this is that it is able to 'compute' by being able to be switched on. So, a switched-off computer isn't a computer (in the meaning of Alan Turing [11, p.230–65]) but only the hardware-condition of the computing potential. With the grammatical form of present progressive ('-ing') retro comput*ing* shows the process of 'present-ating'. For the machine and its user retro computing is a process, not a state. Thus the seeming anachronism of comput*ing* with a vintage home computer (doing something new with something old) dissolves: operative computers contradict any anthropomorphic description (like 'old', 'new', or even 'history') by their operativity.

From the mid-nineties on, consolidation on the computer market did happen. *Commodore's* Amiga 1200, *Atari's* Falcon, and *Acorn's* A3010 (all three were released in 1992) had been the last home computers (following my definition). The end of the 8-bit and 16-bit era was only the end of their production and distribution; the, old 'platforms had been in circulation and this is one of the causes for retrocomputing. Personal computers (PCs) had become very cheap because of the emergence of clones – especially from Asian companies. Their increased audio-visual capabilities, their big amount of RAM and external memories (e.g. CD-ROM drives) and the manifolds of peripherals made them ideal gaming systems.

This is why the computer game industry began concentrating their efforts on Intel CPU based platforms and on gaming consoles. Even the market for those had consolidated in the 1990s. Formerly, big players' like Atari, Coleco, or Sega had gone bankrupt or left the hardware market. Sony with their Playstation (and their successors), Nintendo (N64, Game Cube, DS, Wii, . . .), and Microsoft (Xbox, Xbox 360, Xbox One) competed for console buyers. Home computing started restraining on public domain games and demo programming for PC platforms.

It seems to be the usability of GUI operating systems that caused this convergence: Microsoft's Windows, Apple's Mac OS and some Linux variants with such GUI OS's were able to divide the market among each other. Even if some companies had tried to compete with alternative concepts (like Transputer and RISC CPUs) Intel's x86 microprocessor architecture became the standard for PCs bit by bit. (Two decades later Achorn began to reconquer with RISC CPUs for mobile devices.)

So, home computing had become personal computing: PCs and Macs entered the internet and converted one medium technology after the other into their operating systems so they could be used to listen to music, watch movies and to receive radio and tv. Programming those systems, of course, had been possible, too. This can be determined by the growing output of the demo scenes [cf. 12, p.249ff]. The technological improvement in performance led to an increase in the emulator scene as well. One after another 8-bit home computer, gaming console, and arcade machine had been virtualised by hobbyists. The motivation for this seemed to be nostalgia for the platforms of their childhood and youth and especially for the games they played back in the day.

The demands for the ‚quality' of emulators changed with the increasing possibilities of the host platforms. Emulators became large group projects where hobbyists all around the world exchanged their ideas, knowledge, and codes with the help of the internet – first they used newsgroups and mailing lists, later internet boards, weblogs, wikis, and social media. Information about vintage computers had been stored in large databases and on FTP servers where everyone could look them up. Because of this, retrocomputing enthusiasts were able to gain any needed information about the system they wanted to emulate. Those databases were not restricted to scanned ‚paperware'. Since emulators mostly work with virtualised data carriers, online storages for vintage software on data tapes, discs, cartridges and other were built, where this content could be downloaded as a specific 'image file' to be loaded into an emulator. Such storages were seldom legal since the copyright owners of the historical software still existed. So the tradition of cracker and swapper scenes of the 1980s somehow continued in this practice.

This privately organised and unsystematically performed collecting of historical information and software is an important source for computer history and preservation until today. Of course, the hardware had not been forgotten, too: because of the mass production of microcomputers that started in the late 1970s a huge amount of such systems had been retained by their former users, stored in cellars and attics and often sold at flea markets and online auctions. This was the foundation for retro-computing scenes in the early 2000s, where different home computer systems had been collected, restaurated/repaired and even exhibited. In 1997 the first 'Vintage Computer Festivals' invited retro computing enthusiasts to show off their treasures to the public. Started in the USA 'Vintage Computer Festivals' now happen in numerous countries all over the world to bring collectors, developers for hardware and software, curators and restaurators together to discuss computer history with operative vintage systems.

Retrocomputing hobbyists soon started new projects for old platforms by using information from online databases, chats, the possibilities emulators entailed (cross-platform development, comfortable debugging methods, virtual external drives, ...) and the possibility of co-working on projects via the internet. New games were developed, hardware extensions were created (e.g. for connecting vintage computers to the internet or to use modern mass memory storages with them), new cases, keyboards, and other plastic parts were made with the use of 3D printers. New books and magazines about historical computers came to the shops and even new home computers, on the basis of historical architectures, had been invented.

Today retrocomputing enthusiasts meet online or at computer festivals and fairs to discuss historical topics and plan future projects. Clubs and societies call for competitions on new and interesting hardware and software projects. For example, the German ABBUC (Atari Bit-Byter User Club) tenders an annual price for the most inventive hardware gadget for Atari home computers and there is another annual competition where BASIC programmers can show off their hacker competences in programming 'Tenliner' games for different 8-bit home computers. The following example 'catrpilr' (from the 2019 contest) for the BBC Micro shows the elaborated competence in BASIC hacking:

```
0 k=31:V.278;23;8202;0;0;0;787;6;0;29,435;339;3356;k,25,9:d=32:g=0:h=0:w=0:x=&404
1 f=g>h:h=h-f*(g-h):g=0:m=42:n=4:r=5:z=380:y=x+m:V.k,3,6:C.1:IFf P."HISCORE":w=z
2 n=n-1:a=18:b=13:s=1:TI.=0:IFn<3SO.0,−9,7,9:V.k,n,b,d,k,x?u,y?u,35EL.V.k;b,m,m,m
3 REP.U.TI.>w:V.a;6275;-b;-b;z+b;z+b;6160;4;4;z;z;a;4224;:Q=-x:?x=6:?y=6:IFn=0G.1
4 IFs=1X=0:Y=1:u=0 EL.g=g+5/5^q:r=r-(r<m):F.i=0TO2:SO.0,−8,1A.i,1:N.:IFp q=−1:Q=T
5 p=a=b:REP.b=a:a=RND(11):U.PO.d*a,384-d*b)=0:V.k,a,b,36-p,k,3,13:C.2:P.g:U=TI.
6 j=1A.u:REP.i=INKEY-104-INKEY-103:IFi X=i:Y=0EL.i=INKEY-67-INKEY-82:IFi Y=i:X=0
7 T=TI.:q=T-Q<x:U.T>U+q+5A.T>U+40-r+10*q:U=T:SO.0,−3,j+1,1:v=(u+1)MODm:x?v=x?
   u-X
8 C.j+2:V.787;6+j*q;0;k:y?v=y?u-Y:c=PO.d*x?v,384-d*y?v):t=(v-r+m)MODm:f=r=s:w=120
9 V.x?u,y?u,64,17;k,x?t,y?t,-f*d,273;k,x?v,y?v,w-d*j:u=v:s=s+f+1:ONc+2G.2,6,4,2,2
```

Figure 11 Of course, assembly programmers are using the advanced possibilities for the development of new games, too. Programming assembly language for a historical platform today is easier than ever: with the help of specific integrated development environments (IDEs – for programming and debugging), emulators (for testing), modern SD-card mass memory devices (to transfer programmes from the modern PC to the vintage home computer), and of course with the online-help of countless enthusiasts everybody can learn assembly and master the programming of a game. (See Figure 12)

The benefit of retro computing lies not only in gaming. Since the complexity of contemporary hardware (computers, gaming consoles, mobile devices) and software is way too high to be comprehensible for (and to be programmed by) a non-professional, the likewise easy-to-understand 8-bit and 16-bit computers remain understandable. And since they base on the very same ideas (as Turing machines implemented as a Von-Neumann architecture) as modern digital computers, they can be used for learning how computers work.

This comparability has been used within the last decade for the development of new single-board computers. Such systems were used for teaching purposes back in the 1970s and 1980s to show the

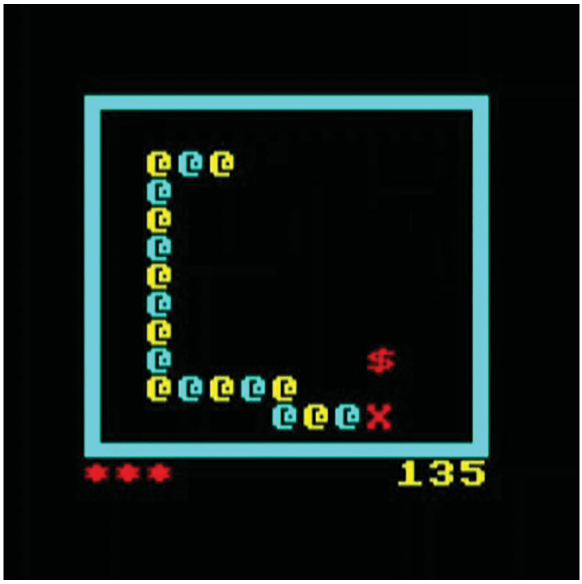

Figure 11. Screenshot from 'catrpilr'.

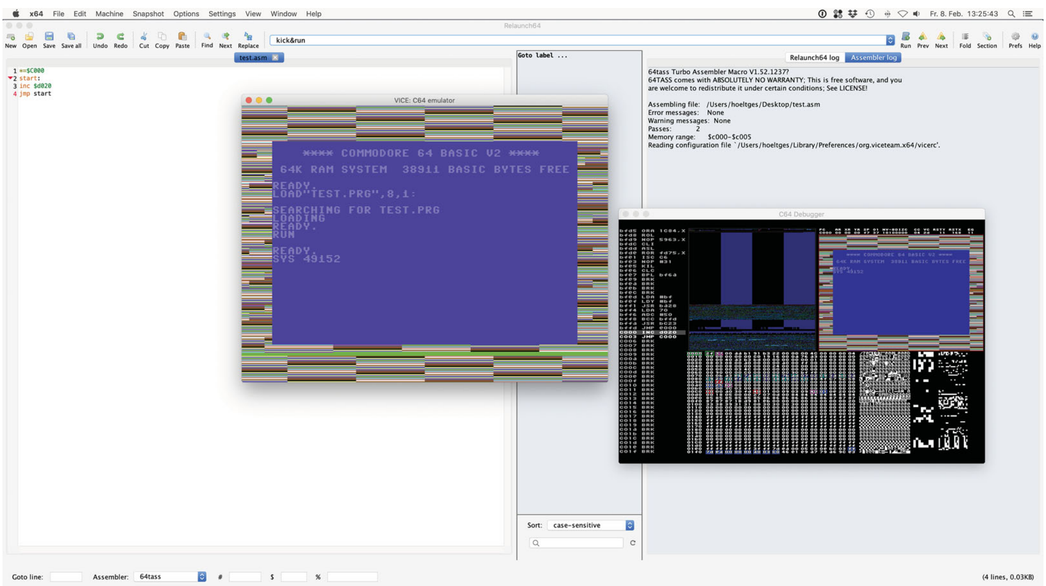

Figure 12. Modern software development for the C64: Relaunch64 (IDE), C64Debugger (Debugger), and VICE (emulator). (The screenshot shows the running programme from Figure 5.)

functioning of hardware and learn to programme (mostly with assembly languages). The electronic basis of computer technology had been visible (and measurable) with those systems for their users all of the time. Modern single-board computers like Arduino, Raspberry Pi, Calliope Mini, or the BBC Micro-Bit (just to name a few of the many available systems) do quote this tradition and are used within school and self-education – and in the hacker context to play with them either with emulators (developed for

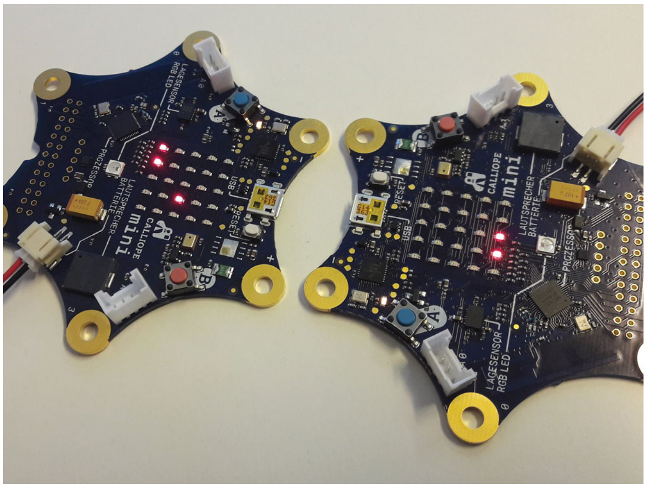

Figure 13. 'Funk-Pong' (wireless Pong) for two BBC MicroBits.

those platforms) or with homebrew games that have to handle the technical limitations of the small computers – in terms of retrocomputing. (See Figure 13)

Acknowledgements

Thank you to Chiara Rochlitz for copy editing this chapter.

Notes

1. However, a short definition of the term home computer should be given: Home computers are electronic digital computers that were built to adapt with already existing private media technology (tv sets, stereo amplifiers, tape recorders, gaming console joysticks), so the acquisition of expensive peripheral hardware could be avoided for private users. For personal computers this peripheral hardware had to be acquired or were implemented (like in the early Apple Macintosh, the Commodore PET 2001 or the Sharp MZ-80) which made them expensive. Another kind of 'homification' of computers can be found in their designs – by using wood or wood imitates for their housings.
2. http://en.wikipedia.org/wiki/E.S.R.,_Inc. (last viewed 2019 March 10).
3. TTL = transistor-transistor logics – integrated circuits that consist of different logical gates that could be coupled to built a computer architecture. [cf. 3, p.103–117].
4. https://en.wikipedia.org/wiki/Honeywell_316 (last viewed 2019 March 10).
5. With time sharing systems that were invented in the early 1960s more than one user can use a computer at the same time. The operating systems splits the resources of calculating time and memory for the attached terminals. This leads to a (seemingly) multi-user/multi-tasking operation of the computer.
6. https://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists (last viewed 2019 March 10).
7. http://www.woz.org/letters/apple-basic/ (last viewed 2019 March 10).
8. https://github.com/cmosher01/Apple-II-Source/blob/master/doc/breakout.asciidoc (last viewed 2019 March 10).
9. A BASIC command like PRINT isn't stored as the text 'PRINT' in memory but as a token (a number). This method saves memory space but it demands time for coding and decoding the token.
10. An opcode is the number for a specific machine language command that is implemented 'hard-wired' into the CPU.
11. The DATA table contains the opcodes for the following assembly language programme:

```
1000: LDA #211; ASCII of the heart-shaped character
1002: LDX #0; reset counter X
1004: STA $8000,X; screen RAM top
1007: STA $8100,X ; screen RAM middle
1010: STA $8200,X; screen RAM bottom
1013: INX; X ← X+11014: CPX #$FF; if X < 255 . . .
1016: BNE $f2 ; . . . then jump back to 1004
1019: RET ; . . . else end
```

12. https://www.youtube.com/watch?v = lUznLFDAzto (last viewed 2019 March 10).
13. *Texas Instruments* introduced an early 16-bit home computer in 1981: the TI-99/4. But this machine had so much constraints in speed and usage it even could not compete with contemporary 8-bit platforms.
14. http://jafma.net/software/nutria/ (last viewed 2019 March 10).
15. http://kaluszka.com/vt/emulation/history.html (last viewed 2019 March 10).
16. *Micro-Soft's* BASIC for the Altair 8800 computer was developed on a PDP-10 mini computer for which Paul Allen and Bill Gates first of all had to programme an Intel 8080 CPU emulator.

Disclosure statement

No potential conflict of interest was reported by the author.

References

[1] Levy S. Hackers: heroes of the computer revolution. Garden City/New York: Anchor Press Doubleday; 1984.
[2] Höltgen S. Data, Dating, Datamining. Der Computer als Medium zwischen Mann und Frau – innerhalb und außerhalb von Fiktionen. In: Bukow GC, Fromme J, Jörissen B, editors. Raum, Zeit, Medienbildung. Untersuchungen zu medialen Veränderungen unseres Verhältnisses zum Raum und Zeit. Wiesbaden: Springer VS; 2012. p. 265–294.
[3] Höltgen, S. Logik für Medienwissenschaftler. In Ders, editor. Medientechnisches Wissen – Band 1: Logik, Informations- und Speichertheorie. Berlin/Boston: De Gruyter; 2017. p. 14–149.
[4] Höltgen S. All Watched Over by Machines of Loving Grace" – Öffentliche Erinnerung, demokratische Informationen und restriktive Technologien am Beispiel der "Community Memory. In: R. Reichert, editor. Big Data. Analysen zum digitalen Wandel von Wissen, Macht und Ökonomie. Bielefeld: transcript; 2014. p. 385–404.
[5] Höltgen, S. Play that pokey music: computer archeological gaming with vintage sound chips. Comp Games J. 2018;7(4) Special Edition 2018 – Ludomusicology, S: 213–230. doi:10.1007/s40869-018-0068-5
[6] Ahl DH. DEC: 101 BASIC computer games. Maynard: Digital Equipment Corporation; 1973.
[7] Ahl DH. BASIC computer games – microcomputer edition. New York: Workman Publishing; 1978.
[8] Ahl DH. More BASIC computer games. New York: Workman Publishing; 1979.
[9] Nelson, T. Computer lib – dream machines (Reprint 1987). Redmont: Microsoft; 1987.
[10] Othmer T. BASIC-Erweiterungen für Spielkonsolen. In: Retro Nr. 29 (Winter 2013/14); p. 34–37.
[11] Turing AM. On computable numbers, with an application to the entscheidungsproblem. Proceedings London Mathematical Society 1937;s2-42(1):230–265.
[12] Botz D. Kunst, Code und Maschine. Die Ästhetik der Computer-Demoszene. Bielefeld: transcript; 2011.